

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-92662

(P2001-92662A)

(43) 公開日 平成13年4月6日 (2001.4.6)

(51) IntCl.⁷G 0 6 F 9/38
9/30

識別記号

3 7 0
3 1 0

F I

G 0 6 F 9/38
9/30

テーマコード* (参考)

3 7 0 C 5 B 0 1 3
3 1 0 E 5 B 0 3 3

審査請求 未請求 請求項の数 8 O L (全 12 頁)

(21) 出願番号 特願平11-269532

(22) 出願日 平成11年9月22日 (1999.9.22)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 宮森 高

神奈川県川崎市幸区小向東芝町1番地 株
式会社東芝マイクロエレクトロニクスセン
ター内

(74) 代理人 100083806

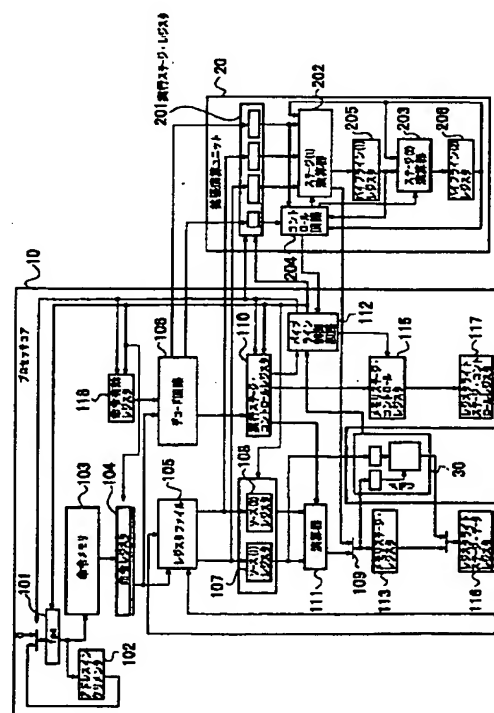
弁理士 三好 秀和 (外7名)

Fターム (参考) 5B013 DD03 EE02

5B033 AA13 AA14 BA03 BC06

(54) 【発明の名称】 プロセッサコア及びこれを用いたプロセッサ

(57) 【要約】

【課題】 外部に演算ユニットを効率よく接続することが
できるプロセッサコアを提供する。【解決手段】 外部に接続された拡張演算ユニットを制
御するための命令である拡張演算ユニット制御命令を実
装したプロセッサコアであって、命令をデコードする機
能を有し、そのデコードした命令が前記拡張演算ユニッ
ト制御命令であった場合には、前記拡張演算ユニットに
対して、演算データ、拡張演算コードの動作を規定する
命令コード、及び該命令コードが有効であることを示す
情報を出力する命令デコード部を備える。すなわち、プ
ロセッサコアは、拡張演算ユニットを制御するための拡
張命令を備え、さらに拡張命令を実行するための制御手
段と、拡張演算ユニットを制御しデータを転送するイン
ターフェース手段とを内蔵した。

BEST AVAILABLE COPY

【特許請求の範囲】

【請求項 1】 外部に接続される拡張演算ユニットを制御するための命令である拡張演算ユニット制御命令を実装したプロセッサコアであって、命令をデコードする機能を有し、そのデコードした命令が前記拡張演算ユニット制御命令であった場合には、前記拡張演算ユニットに対して、演算データ、拡張演算コードの動作を規定する命令コード、及び該命令コードが有効であることを示す情報を出力する命令デコード部を備えたことを特徴とするプロセッサコア。

【請求項 2】 前記拡張演算ユニットに対して出力する前記演算データは、該拡張演算ユニットを制御する命令の一部で指定されたレジスタ番号に従って、プロセッサ内部のレジスタファイルから読み出された値であることを特徴とする請求項 1 記載のプロセッサコア。

【請求項 3】 パイプライン停止信号を前記拡張演算ユニットへ出力して、該パイプライン停止信号が有効なときに、前記拡張演算ユニットの実行を一時停止する手段を有することを特徴とする請求項 1 記載のプロセッサコア。

【請求項 4】 パイプラインフラッシュ信号を前記拡張演算ユニットへ出力して、該パイプラインフラッシュ信号が有効なときに、前記拡張演算ユニットへ出力した命令の実行を破棄する手段を有することを特徴とする請求項 1 記載のプロセッサコア。

【請求項 5】 前記拡張演算ユニットからパイプライン停止信号を入力し、該パイプライン停止信号が有効なときに、命令の実行を一時停止する手段を有することを特徴とする請求項 1 記載のプロセッサコア。

【請求項 6】 複数の演算器を有し、前段の演算器の処理結果を次のクロックの立上りで格納する第 1 のパイプラインレジスタと、命令の実行をパイプライン的に処理する第 2 のパイプラインレジスタとを有する拡張演算ユニットと、命令をデコードする機能を有し、デコードした命令が前記拡張演算ユニットを制御する命令であった場合には、前記拡張演算ユニットに対して、演算データ、拡張演算コードの動作を規定する命令コード、及び該命令コードが有効であることを示す情報を出力する命令デコード部を有するプロセッサコアとを備えたことを特徴とするプロセッサ。

【請求項 7】 前記拡張演算ユニットから、アドレス、データ、データ書き込みを制御する書き込み制御信号、及びデータ読み出しを制御する読み出し制御信号を入力し、前記読み出し制御信号が有効でデータ読み出しを行う時には、前記アドレスで指定された領域からデータを読み出して前記拡張演算ユニットへ出力し、前記書き込み制御信号が有効でデータ書き込みを行う場合には、前記アドレスで指定された領域に、前記拡張演算ユニット

からのデータを書き込むメモリを有することを特徴とする請求項 6 記載のプロセッサ。

【請求項 8】 前記メモリは、前記プロセッサコア及び前記拡張演算ユニットからアクセス可能に構成したことを特徴とする請求項 7 記載のプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、命令を実行するプロセッサコア、及びこれを用いたプロセッサに関する。

【0002】

【従来の技術】従来、この種の分野の技術に関しては、例えば図 7 に示すようなものがあった。

【0003】図 7 は、従来のプロセッサの基本構成を示すブロック図である。

【0004】このプロセッサは、プロセッサコア 70 とデータメモリ 80 とを備えている。

【0005】プロセッサコア 70 は、フェッチ p C (f P e) 701、アドレスインクリメント 702、命令メモリ 703、命令レジスタ 704、レジスタファイル 705、デコード回路 706、ソース (1) レジスタ 707、ソース (2) レジスタ 708、実行ステージ・コントロールレジスタ 710、演算器 711、パイプライン制御回路 712、メモリステージ・データレジスタ 713、メモリステージ・コントロールレジスタ 715、レジスタライトステージ・レジスタ 716、レジスタライトステージ・コントロールレジスタ 717、及び命令有効レジスタ 718 から構成されている。

【0006】プロセッサコア 70 は、5 段のパイプライン構成で命令を実行する。すなわち、パイプラインステージ 1 (命令フェッチステージ) と、パイプラインステージ 2 (レジスタリードステージ) と、パイプラインステージ 3 (実行ステージ) と、パイプラインステージ 4 (メモリステージ) と、パイプラインステージ 5 (レジスタライトステージ) とを有する。

【0007】ここで、命令フェッチステージは、命令メモリ 703 から命令を読み出すステージであり、レジスタリードステージは、命令フェッチステージで読み出した命令で指定された、レジスタファイル 705 中のレジスタを読み出すとともに、デコード回路 706 で命令をデコードするステージである。実行ステージは、レジスタリードステージで読み出したレジスタの値と命令のデコード情報によって、演算器 711 で命令を実行するステージであり、メモリステージは、メモリロード、ストア命令の場合にはデータメモリ 80 をアクセスする。レジスタライトステージは、実行ステージでの実行結果あるいはメモリロード命令の場合のロードデータをレジスタファイル 705 へ書き込む。

【0008】

【発明が解決しようとする課題】しかしながら、従来のプロセッサコアでは、予め定義された命令を実行するだ

けで、外部に演算ユニットを接続するための拡張機能がなかった。すなわち、外部に演算ユニットを効率よく接続するための機構やインターフェース信号がないため、応用システムに適した演算ユニットを効率よくプロセッサコアに接続して、システム性能の向上を図ることができない。

【0009】そこで、応用システムに適した演算ユニット、例えば積和演算器をプロセッサコアに予め内蔵することも考えられるが、応用システムによっては、その積和演算器を全く実行しないものもある。従って、プロセッサコアに一樣に前記積和演算器を内蔵することは無駄なハードウェアとなり、コスト増大の要因となる。

【0010】このようなことから、従来のプロセッサコアでは、コア外部に演算ユニットを効率よく接続するための拡張機能が求められていた。

【0011】本発明は、上述の如き従来の問題点を解決するためになされたもので、その目的は、外部に演算ユニットを効率よく接続することができるプロセッサコアと、これを用いたプロセッサを提供することである。

【0012】

【課題を解決するための手段】上記目的を達成するために、請求項1記載の発明に係るプロセッサコアでは、外部に接続される拡張演算ユニットを制御するための命令である拡張演算ユニット制御命令を実装したプロセッサコアであって、命令をデコードする機能を有し、そのデコードした命令が前記拡張演算ユニット制御命令であった場合には、前記拡張演算ユニットに対して、演算データ、拡張演算コードの動作を規定する命令コード、及び該命令コードが有効であることを示す情報を出力する命令デコード部を備えたことを特徴とする。

【0013】請求項2記載の発明に係るプロセッサコアでは、請求項1記載のプロセッサコアにおいて、前記拡張演算ユニットに対して出力する前記演算データは、該拡張演算ユニットを制御する命令の一部で指定されたレジスタ番号に従って、プロセッサ内部のレジスタファイルから読み出された値であることを特徴とする。

【0014】請求項3記載の発明に係るプロセッサコアでは、請求項1記載のプロセッサコアにおいて、パイプライン停止信号を前記拡張演算ユニットへ出力して、該パイプライン停止信号が有効なときに、前記拡張演算ユニットの実行を一時停止する手段を有することを特徴とする。

【0015】請求項4記載の発明に係るプロセッサコアでは、請求項1記載のプロセッサコアにおいて、パイプラインフラッシュ信号を前記拡張演算ユニットへ出力して、該パイプラインフラッシュ信号が有効なときに、前記拡張演算ユニットへ出力した命令の実行を破棄する手段を有することを特徴とする。

【0016】請求項5記載の発明に係るプロセッサコアでは、請求項1記載のプロセッサコアにおいて、前記拡張

演算ユニットからパイプライン停止信号を入力し、該パイプライン停止信号が有効なときに、命令の実行を一時停止する手段を有することを特徴とする。

【0017】請求項6記載の発明に係るプロセッサでは、複数の演算器を有し、前段の演算器の処理結果を次のクロックの立上りで格納する第1のパイプラインレジスタと、命令の実行をパイプライン的に処理する第2のパイプラインレジスタとを有する拡張演算ユニットと、命令をデコードする機能を有し、デコードした命令が前記拡張演算ユニットを制御する命令であった場合には、前記拡張演算ユニットに対して、演算データ、拡張演算コードの動作を規定する命令コード、及び該命令コードが有効であることを示す情報を出力する命令デコード部を有するプロセッサコアとを備えたことを特徴とする。

【0018】請求項7記載の発明に係るプロセッサでは、請求項6記載のプロセッサにおいて、前記拡張演算ユニットから、アドレス、データ、データ書き込みを制御する書き込み制御信号、及びデータ読み出しを制御する読み出し制御信号を入力し、前記読み出し制御信号が有効でデータ読み出しを行う時には、前記アドレスで指定された領域からデータを読み出して前記拡張演算ユニットへ出力し、前記書き込み制御信号が有効でデータ書き込みを行う場合には、前記アドレスで指定された領域に、前記拡張演算ユニットからのデータを書き込むメモリを有することを特徴とする。

【0019】請求項8記載の発明に係るプロセッサでは、請求項7記載のプロセッサにおいて、前記メモリは、前記プロセッサコア及び前記拡張演算ユニットからアクセス可能に構成したことを特徴とする。

【0020】

【発明の実施の形態】以下、本発明の実施の形態を図面に基づいて説明する。

【0021】[第1実施形態] 図1は、本発明の第1実施形態に係るプロセッサの構成を示すブロック図である。

【0022】本実施形態のプロセッサは、プロセッサコア10と、このプロセッサコア10の外部に接続された拡張演算ユニット20と、データメモリ30と備えている。

【0023】初めに、プロセッサコア10の説明を行う。

【0024】このプロセッサコア10は、フェッチPC(fPc)101、アドレスインクリメント102、命令メモリ103、命令レジスタ104、レジスタファイル105、デコード回路106、ソース(1)レジスタ107、ソース(2)レジスタ108、実行ステージ・コントロールレジスタ110、演算器111、パイプライン制御回路112、メモリステージ・データレジスタ113、メモリステージ・コントロールレジスタ115、レジスタライトステージ・データレジスタ116、

10

20

30

40

50

レジスタライトステージ・コントロールレジスタ117、及び命令有効レジスタ118から構成されている。

【0025】プロセッサコア10は、次の5段のパイプライン構成で命令を実行する。すなわち、パイプラインステージとして、命令メモリ103から命令を読み出すステージであるパイプラインステージ1（命令フェッチステージ）と、命令フェッチステージで読み出した命令で指定された、レジスタファイル105中のレジスタを読み出すとともに、デコード回路106で命令をデコードするパイプラインステージ2（レジスタリードステージ）と、レジスタリードステージで読み出したレジスタの値と命令のデコード情報とによって、演算器111で命令を実行するパイプラインステージ3（実行ステージ）と、メモリロードやストア命令の場合にデータメモリ30をアクセスするパイプラインステージ4（メモリロードステージ）と、実行ステージでの実行結果あるいはメモリロード命令の場合のロードデータをレジスタファイル105へ書き込むパイプラインステージ5（レジスタライトステージ）とを有する。

【0026】本実施形態のプロセッサコア10の代表的な命令としては、例えば、加算命令（add）、データメモリロード命令（load）、データメモリストア命令（store）、ジャンプ命令（jump）、及び拡張命令（ext）があり、それぞれの命令は32ビット長とする。

【0027】図2は、これら各命令のビットパターンを示す図であり、同図を参照して各命令の動作を説明する。

【0028】加算命令（add）は、32ビット命令のビット19～16が0000であり、src1フィールドで指定されるレジスタとsrc2フィールドで指定されるレジスタの値とを加算して、dstフィールドで指定されるレジスタへ書き込む。

【0029】データメモリロード命令（load）は、命令のビット19～16が0001であり、命令の動作は、src1フィールドで指定されるレジスタの値をデータメモリ30のアドレスとして、データメモリ30のデータを読み出し、dstフィールドで指定されるレジスタへ書き込む。

【0030】データメモリストア命令（store）は、命令のビット19～16が0010であり、命令の動作は、src1フィールドで指定されるレジスタの値をデータメモリ30のアドレスとして、src2フィールドで指定されるレジスタの値をデータメモリ30へ書き込む。

【0031】ジャンプ命令（jump）は、命令のビット19～16が0011であり、src1フィールドで指定されるレジスタの値を読み出し、ジャンプ先アドレスとしてジャンプを行う。

【0032】拡張命令（ext）は、命令のビット19

～16が0100であり、src1フィールドで指定されるレジスタとsrc2フィールドで指定されるレジスタの値とを読み出し、外部の拡張演算ユニット20に出力する。拡張演算ユニット20では演算結果をプロセッサコア10へ出力し、その結果をdstフィールドで指定されるレジスタへ書き込む。

【0033】上述した各パイプラインステージの基本的な動作を説明する。

【0034】〈パイプラインステージ1の動作〉パイプラインステージ1（命令フェッチステージ）では、フェッチpc101の値により命令メモリ103から命令を読み出す。次のクロックの立上りで、命令メモリ103から読み出された命令が命令レジスタ104へ書き込まれるとともに、命令有効レジスタ118の値は1にセットされる。また、フェッチPC101の値がアドレスインクリメンタ102で命令ワードのバイト数（命令ワードが4バイトの場合は4）だけ加えられ、次のクロックの立上りでフェッチpc101に格納され、次の命令が命令メモリ103から読み出されるという動作を繰り返す。

【0035】〈パイプラインステージ2の動作〉パイプラインステージ2（レジスタリードステージ）では、命令レジスタ104の命令で指定されたレジスタの値をレジスタファイル105から読み出す。図2に示した命令の場合では、命令のビット27～24がsrc1フィールド、ビット23～20がsrc2フィールドであり、このビットで指定されたレジスタの値がレジスタファイル105に読み出される。レジスタファイル105から読み出されたレジスタの値は、次のクロックの立上りエッジで、ソース（1）レジスタ107及びソース（2）レジスタ108に格納される。

【0036】また、デコード回路106は、命令有効レジスタ118の値が1で命令レジスタ104に有効な命令が格納されている場合に、命令レジスタ104の命令をデコードして実行ステージ・コントロールレジスタ110へ、演算器111の制御情報と、デスティネーションレジスタ番号及びその有効情報と、ジャンプ命令によるパイプラインフラッシュ情報とを出力する。これらの情報は、次のクロックの立上りで、実行ステージ・コントロールレジスタ110に格納される。

【0037】さらに、デコード回路106の動作を詳細に説明すると、デコード回路106は、命令レジスタ104から32ビットの命令を読み出し、図3に示すように制御情報を出力する。デスティネーションレジスタ番号（dst番号）としては命令のビット31～28を出力する。デスティネーション有効情報は、命令実行の結果、レジスタへの書き込みを行うadd, load, ext命令の場合には1（有効）となる。パイプラインフラッシュ情報（Flush）はジャンプ命令のとき1（有効）となる。拡張命令有効情報（ext有効）はe

ext命令の場合に1（有効）となる。また、拡張命令情報（code16）としては、命令コードのビット15～0が出力される。

【0038】〈パイプラインステージ3の動作〉パイプラインステージ3（実行ステージ）では、演算器111はソース（1）レジスタ107及びソース（2）レジスタ108からデータを入力して、実行ステージ・コントロールレジスタ110内の、演算器111の制御情報に従って演算を行う。演算器111の制御情報は、図3に示したデコード回路106の出力に対応しており、演算器111は図4に示すような動作を行う。

【0039】演算器111の演算結果は、メモリステージ・データレジスタ113に格納されるとともに、データメモリ30をアクセスするデータロード命令とデータストア命令の場合では演算器111の出力をデータメモリ30のアドレスとして出力する。データストア命令の場合は、セクタ109からの値をデータメモリ30へストアするデータとして出力する。

【0040】実行ステージ・コントロールレジスタ110のデスティネーションレジスタ番号とその有効情報が、次のクロックの立上りでメモリステージ・コントロールレジスタ115へ格納される。

【0041】実行ステージ・コントロールレジスタ110にジャンプ命令によるパイプラインフラッシュ情報が有効になった場合は、パイプライン制御回路112ではパイプラインフラッシュ信号を有効にして、次のクロックの立上りで、命令有効レジスタ118及び実行ステージ・コントロールレジスタ110の有効情報を無効にして、次のクロックでのデコードステージ及び実行ステージが動作しないように制御する。

【0042】また、パイプラインフラッシュ信号がアクティブになると、演算器111の出力のジャンプ先アドレスが次のフェッチpc101の値になるように制御する。

【0043】同時に、パイプラインフラッシュ信号を拡張演算ユニット20へ出力し、拡張演算ユニット20内部の実行ステージレジスタ201を無効にできるようにする。

【0044】パイプライン制御回路112は、データメモリ30からのパイプラインストール（停止）要求、あるいは拡張演算ユニット20からのパイプライン停止要求が発生した場合に、フェッチpc101、命令レジスタ104、命令有効レジスタ118、ソース（1）レジスタ107、ソース（2）レジスタ108、及び実行ステージ・コントロールレジスタ110の値が、次のクロックの立上りで更新されないように制御信号を各モジュールへ出力する。

【0045】データメモリ30からのパイプラインストール要求の場合は、メモリステージ・コントロールレジスタ115の値も更新されないように、メモリステージ

・コントロールレジスタ115へ制御信号を出力する。同時に、パイプライン制御回路112は、拡張演算ユニット20へパイプライン停止信号を出力し、拡張演算ユニット20内部の実行ステージレジスタ201の値が更新されないように制御する。

【0046】〈パイプラインステージ4の動作〉パイプラインステージ4（メモリステージ）では、データロード命令の場合はデータメモリ30からデータが読み出され、次のクロックでレジスタライトステージ・データレジスタ116に格納される。その他の命令の場合は、メモリステージ・データレジスタ113の値が、次のクロックでレジスタライトステージ・データレジスタ116に格納される。また、メモリステージ・コントロールレジスタ115の情報が次のクロックで、レジスタライトステージ・コントロールレジスタ117に格納される。

【0047】〈パイプラインステージ5の動作〉パイプラインステージ5（レジスタライトステージ）では、レジスタライトステージ・コントロールレジスタ117のデスティネーションレジスタへの書き込みが有効の場合は、レジスタライトステージ・コントロールレジスタ117内のデスティネーションレジスタ番号で指定されるレジスタファイル105のレジスタへ、レジスタライトステージ・データレジスタ116の値を書き込むように制御する。

【0048】次に、本実施形態に係る拡張演算ユニット20について説明する。

【0049】拡張演算ユニット20は、プロセッサコア10と接続されており、プロセッサコア10のデコードステージで、コア10から、レジスタファイル105より読み出した命令のsrc1、src2フィールドに対応するレジスタの値を入力する。

【0050】デコード回路106から、命令の下位16ビットに相当する拡張命令情報（code16）を入力すると共に、デコードステージの命令が拡張命令extの場合に有効（1）になる拡張命令有効情報を入力する。これらの情報は、コア10からのパイプラインフラッシュ信号、あるいはパイプラインストール信号が有効にならない場合は、次のクロックの立上りで、実行ステージレジスタ201へ格納される。プロセッサコア10からのパイプラインストール信号が有効なときは、実行ステージレジスタ201の値を更新しない。

【0051】プロセッサコア10からのパイプラインフラッシュ信号が有効なときは、次のクロックの立上りで、拡張命令有効情報を強制的に無効（0）にして、拡張演算ユニット20が演算を行わないように制御する。

【0052】ステージ（1）演算器202は、プロセッサコア10のメモリステージに対応したステージで演算を行うユニットである。このステージ（1）演算器202の動作は、コア10から入力した拡張命令情報code16をコントロール回路204がデコードして行う。

またステージ（１）演算器２０２は、拡張命令情報の一部を演算の入力データとして使用することもできる。

【００５３】ステージ（１）の演算結果は、次のクロックの立上りでパイプライン（１）レジスタ２０５に格納される。パイプライン（１）レジスタ２０５の結果は、次のステージ（２）演算器２０３の入力データとして使うことができる。ステージ（２）演算器２０４の結果は、次のクロックの立上りで、パイプライン（２）レジスタ２０６に格納される。

【００５４】本実施形態での拡張命令の例を以下に挙げる。

```

【００５５】 1ext dst,src1,src2,0x0001 acc <- rc1
               *src2;dst <- src1
2ext dst,src1,src2,0x0002 acc <- acc+src1*src2;dst
               <- src1
3ext dst,src1,src2,0x0003 dst <- acc

```

これらの命令を実行するため、拡張演算ユニット２０のステージ（１）演算器２０２は乗算を行い、ステージ（２）演算器２０３は、パイプライン（１）レジスタ２０５から入力したステージ（１）演算器２０２の乗算の結果とパイプライン（２）レジスタ２０６とを加算する。ステージ（２）演算器２０３の結果は、パイプライン（２）レジスタ２０６（この場合はaccレジスタ）に再度格納される。

【００５６】上記１の命令では、プロセッサコア１０から入力したソース（１）データとソース（２）データを入力として、ステージ（１）演算器２０２で乗算を行う。結果は、次のクロックの立上りでパイプライン

（１）レジスタ２０５に格納される。ステージ（２）演算器２０３では、処理を行わず、パイプライン（１）レジスタ２０５に格納された乗算結果を次のクロックの立上りでパイプライン（２）レジスタ２０６に格納する。また、ステージ（１）演算器２０２から、ソース（１）データを命令の結果としてプロセッサコア１０へ出力する。これは、プロセッサコア１０の実行ステージに相当し、プロセッサコア１０は、上記１の命令の結果として、レジスタライトステージでレジスタファイル１０５へ書き込む。

【００５７】上記２の命令では、ステージ１までは、上記１の命令と同様に動作する。ステージ（２）演算器２０３では、パイプライン（１）レジスタ２０５の結果とパイプライン（２）レジスタ２０６の結果とを加算し、その結果をパイプライン（２）レジスタ２０６へ格納する。また、ステージ（１）演算器２０２から、ソース（１）データを命令の結果としてプロセッサコア１０へ出力する。これは、プロセッサコア１０の実行ステージに相当し、プロセッサコア１０は、上記１の命令の結果として、レジスタライトステージでレジスタファイル１０５へ書き込む。

【００５８】上記３の命令では、パイプライン（２）レ

ジスタ２０６を読み出し、ステージ（１）演算器２０２から、ソース（１）データを命令の結果としてプロセッサコア１０へ出力する。これは、プロセッサコア１０の実行ステージに相当し、プロセッサコア１０は、上記１の命令の結果として、レジスタライトステージでレジスタファイル１０５へ書き込む。

【００５９】パイプライン（２）レジスタ２０６には、上記１あるいは２の命令の結果が格納される。したがって、上記３の命令が実行ステージ・レジスタ２０１に格納されたとき、パイプライン（１）レジスタ２０５に上記１あるいは２の命令が格納されることを、コントロール回路２０４はチェックし、この場合は、１サイクルだけコア１０へのパイプラインストール信号を有効にして、パイプラインを停止させ、次のクロックサイクルで、上記１あるいは２の命令の結果を読み出して、ステージ（１）演算器２０２を介して、コア１０へ結果を出力することができる。

【００６０】上述したように、本実施形態では、プロセッサコア１０に、拡張演算ユニット２０を制御するための拡張命令を備え、この拡張命令を実行するための制御回路と、拡張演算ユニット２０を制御しデータを転送するインターフェース回路とを内蔵したので、プロセッサコア１０の外部に、効率よく拡張演算ユニット２０を接続することができる。

【００６１】例えば、拡張演算ユニット２０として、レイテンシが２サイクルでスループットが１サイクルの積和演算器を設けた場合を考える。プロセッサコア１０の乗算命令では実行に５サイクル、加算命令は１サイクルかかったとする。１０回の積和演算は、拡張演算ユニット２０の積和演算器を用いることにより、１１サイクルで実行することができる。これに対して、プロセッサコア１０の乗算命令と加算命令の組合せによる場合は、 $(5+1) \times 10 = 60$ で、６０サイクルかかることになる。

【００６２】プロセッサコアに予め積和演算器を内蔵することも考えられるが、ある応用システムで積和演算が全く実行されない場合、この応用システムでは積和演算ユニットが無駄なハードウェアとなり、コスト増大の要因となる。

【００６３】本実施形態では、積和演算の例以外にも、応用システムに適した演算ユニットをプロセッサコア１０の外部に接続することによって、システム性能を向上させることが可能となる。

【００６４】なお、本実施形態では、拡張演算ユニット２０はステージ（１）演算器２０２とステージ（２）演算器２０３という２段のパイプライン構成を採ったが、拡張演算ユニット２０の構成はこれに限定されるものではなく、３段以上の多段のパイプラインステージを持つことができる。

【００６５】また、プロセッサコア１０へのパイプライン

ンストール要求の例として、実行に複数ステージが必要な命令の結果を読み出す場合を示したが、これに限定されるものではない。例えば、内部がパイプライン処理されず、命令の実行に複数サイクルが必要な場合は、次の命令の実行を、現在実行している命令が終了するまで待たせることが考えられる。これは、コントロール回路 204 がこのような命令実行のステートを管理することで実現できる。

【0066】〔第2実施形態〕図5は、本発明の第2実施形態に係るプロセッサの構成を示すブロック図であり、図1と共通の要素には、同一の符号を付しその説明を省略する。

【0067】第2実施形態のプロセッサでは、プロセッサコア10の構成は図1と同じであるが、拡張演算ユニット40のステージ(1)演算器402からデータメモリ31へ、アドレス信号、リード/ライトのコントロール信号、及びライトデータ信号が出力され、データメモリ31を直接アクセスできるようになっている。

【0068】データメモリ31は、セクタ31a、31bによって、コア10からのアドレス信号、リード/ライトコントロール信号、及びライトデータ信号を選択して、RAMモジュールへ出力することで、コア10と拡張演算ユニット40からのアクセス要求を調停する。

【0069】データメモリ31から読み出されたデータは、拡張演算ユニット40へ出力され、ステージ(2)演算器403の出力と選択されてパイプライン(2)レジスタ406へ、次のクロックの立上りで格納される。

【0070】本実施形態の拡張演算ユニット40では、ステージ(3)演算器407が、パイプライン(2)レジスタ406の出力を入力データとして演算に使用している。このように、データメモリ31から読み出した結果を拡張演算ユニット40での演算入力データとして使用することができる。

【0071】〔第3実施形態〕図6は、本発明の第3実施形態に係るプロセッサの構成を示すブロック図であり、図1と共通の要素には、同一の符号を付しその説明を省略する。

【0072】本実施形態のプロセッサは、図1に示した構成において、拡張演算ユニット20からデスティネーションレジスタ有効ビットを出力することにより、プロセッサコア10の実行ステージでの演算結果をプロセッサコア10で無効にし、レジスタライトステージでレジスタへ書き込まないようにしたものである。

【0073】本実施形態では、上記第1実施形態で示した拡張命令の例を次のように変更する。

【0074】

```
1 ext src1,src2,0x0001 acc <- src1*src2;
2 ext src1,src2,0x0002 acc <- acc+src1*src2;
3 ext dst,src1,src2,0x0003 dst <- acc
```

上記1及び2の命令では、プロセッサコア10のレジス

タファイル105へ結果を書き戻さないため、拡張演算ユニット20のコントロール回路204aは、デスティネーション有効情報を無効として、プロセッサコア10へ出力する。プロセッサコア10の実行ステージ・コントロールレジスタ110のデスティネーションレジスタの有効情報は、この場合には無効(0)として次のクロックの立上りでメモリステージ・コントロールレジスタ115aへ格納される。

【0075】上記3の命令では、プロセッサコア10のレジスタファイル105へ結果を書き戻す。このため、拡張演算ユニット20のコントロール回路204aは、デスティネーション有効情報を有効として、プロセッサコア10へ出力する。

【0076】プロセッサコア10は、拡張命令を実行している場合に、拡張演算ユニット20のコントロール回路204aから出力されるデスティネーション有効情報を、次のクロックの立ち上がりでメモリステージ・コントロールレジスタ115aに格納するように制御する。

【0077】プロセッサコア10のパイプライン制御によっては、実行ステージとメモリステージのデスティネーションレジスタが有効で、そのレジスタ番号がレジスタリードステージで読み出すレジスタ番号と一致したとき、パイプラインをストールさせる場合がある。拡張命令で、プロセッサコア10のレジスタファイル105への書き込みを行わない命令がある場合には、デスティネーションレジスタへの書き戻しを無効という情報を、拡張演算ユニット20からプロセッサコア10へ出力する機能を設けることによって、本来不要であるパイプラインのストールを回避することができる。

【0078】

【発明の効果】以上詳細に説明したように、本発明よれば、拡張演算ユニットを制御するための命令である拡張演算ユニット命令をサポートし、この拡張演算ユニット命令の実行を制御する機能と、拡張演算ユニットを接続するインターフェース機能とを備えたので、プロセッサコアの外部に、効率よく拡張演算ユニットを接続することが可能になる。これにより、応用システムに適した拡張演算ユニットを、プロセッサコアの外部に接続することによって、システム性能を容易に向上させることが可能となる。

【図面の簡単な説明】

【図1】本発明の第1実施形態に係るプロセッサの構成を示すブロック図である。

【図2】プロセッサコアの命令のビットパターンを示す図である。

【図3】図1に示したデコード回路106の動作を示す図である。

【図4】図1に示した演算器111の動作を示す図である。

【図5】本発明の第2実施形態に係るプロセッサの構成

を示すブロック図である。

【図 6】本発明の第 3 実施形態に係るプロセッサの構成を示すブロック図である。

【図 7】従来のプロセッサの基本構成を示すブロック図である。

【符号の説明】

10 プロセッサコア
20, 40 拡張演算ユニット
30, 31 データメモリ
101 フェッチ PC (f P c)
102 アドレスインクリメンタ
103 命令メモリ
104 命令レジスタ
105 レジスタファイル
106 デコード回路
107 ソース (1) レジスタ

108 ソース (2) レジスタ
110 実行ステージ・コントロールレジスタ
111 演算器
112 パイプライン制御回路
113 メモリステージ・データレジスタ
115 メモリステージ・コントロールレジスタ
116 レジスタライトステージ・データレジスタ
117 レジスタライトステージ・コントロールレジスタ
10 118 命令有効レジスタ
202, 402 ステージ (1) 演算器
203, 403 ステージ (2) 演算器
205, 405 パイプライン (1) レジスタ
206, 406 パイプライン (2) レジスタ
407 ステージ (3) 演算器
408 パイプライン (3) レジスタ

【図 2】

	31	2827	2423	2019	16	15	0
add dst, src1, src2	dst	src1	src2	0000	0000	0000	0000
load dst, (src1)	dst	src1	0000	0001	0000	0000	0000
store (src1), src2	0000	src1	src2	0010	0000	0000	0000
jump src1	0000	src1	0000	0011	0000	0000	0000
ext dst, src1, src2, code16	dst	src1	src2	0100	code16		

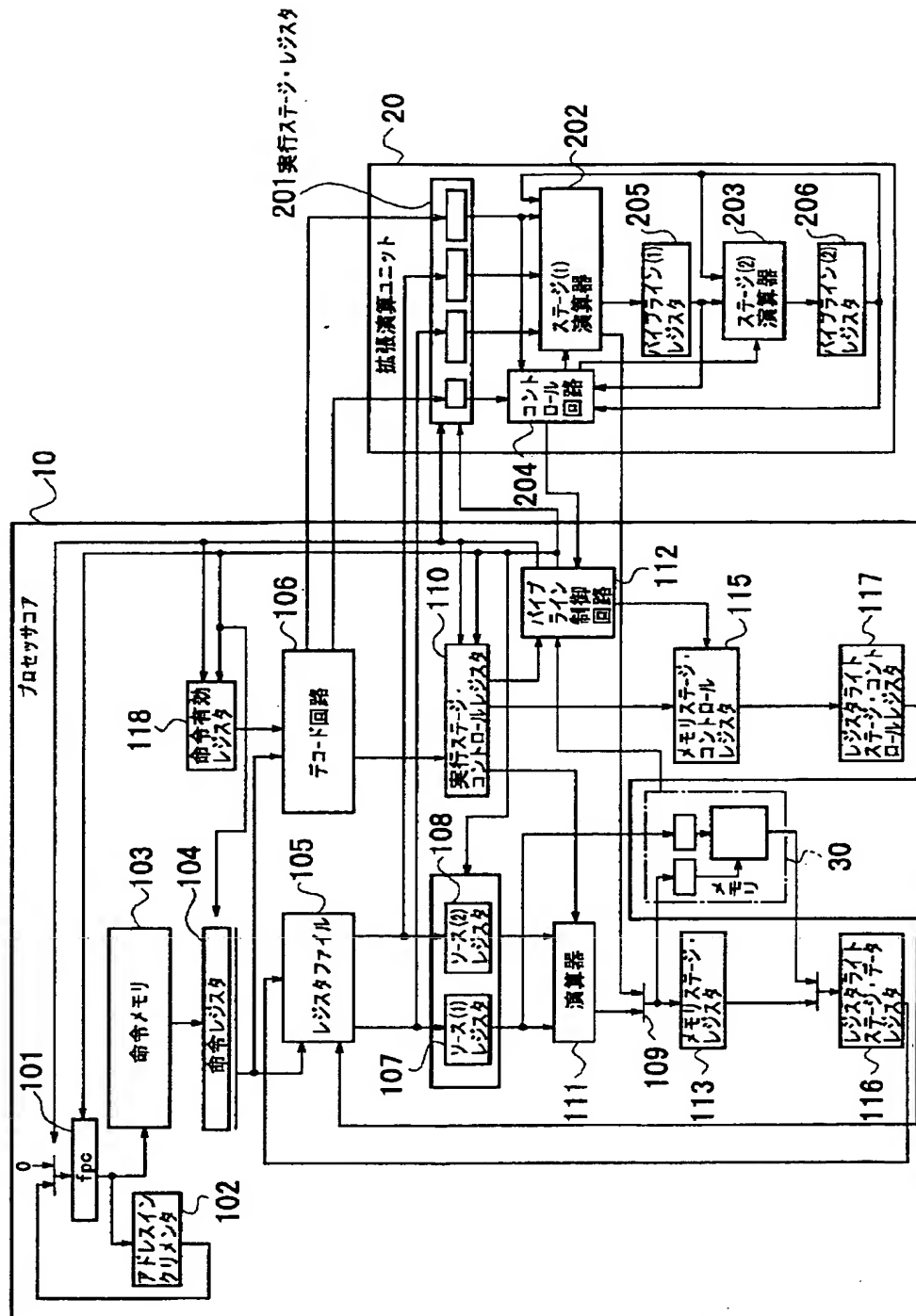
【図 3】

命令入力		出力					
bit	19~16	演算器制御	dst番号	dst有効	Flush	ext有効	code16
add	0000	01(add)	bit31~28	1	0	0	bit15~0
load	0001	00(nop)	bit31~28	1	0	0	bit15~0
store	0010	00(nop)	bit31~28	0	0	0	bit15~0
jump	0011	00(nop)	bit31~28	0	1	0	bit15~0
ext	0100	10(ext)	bit31~28	1	0	1	bit15~0

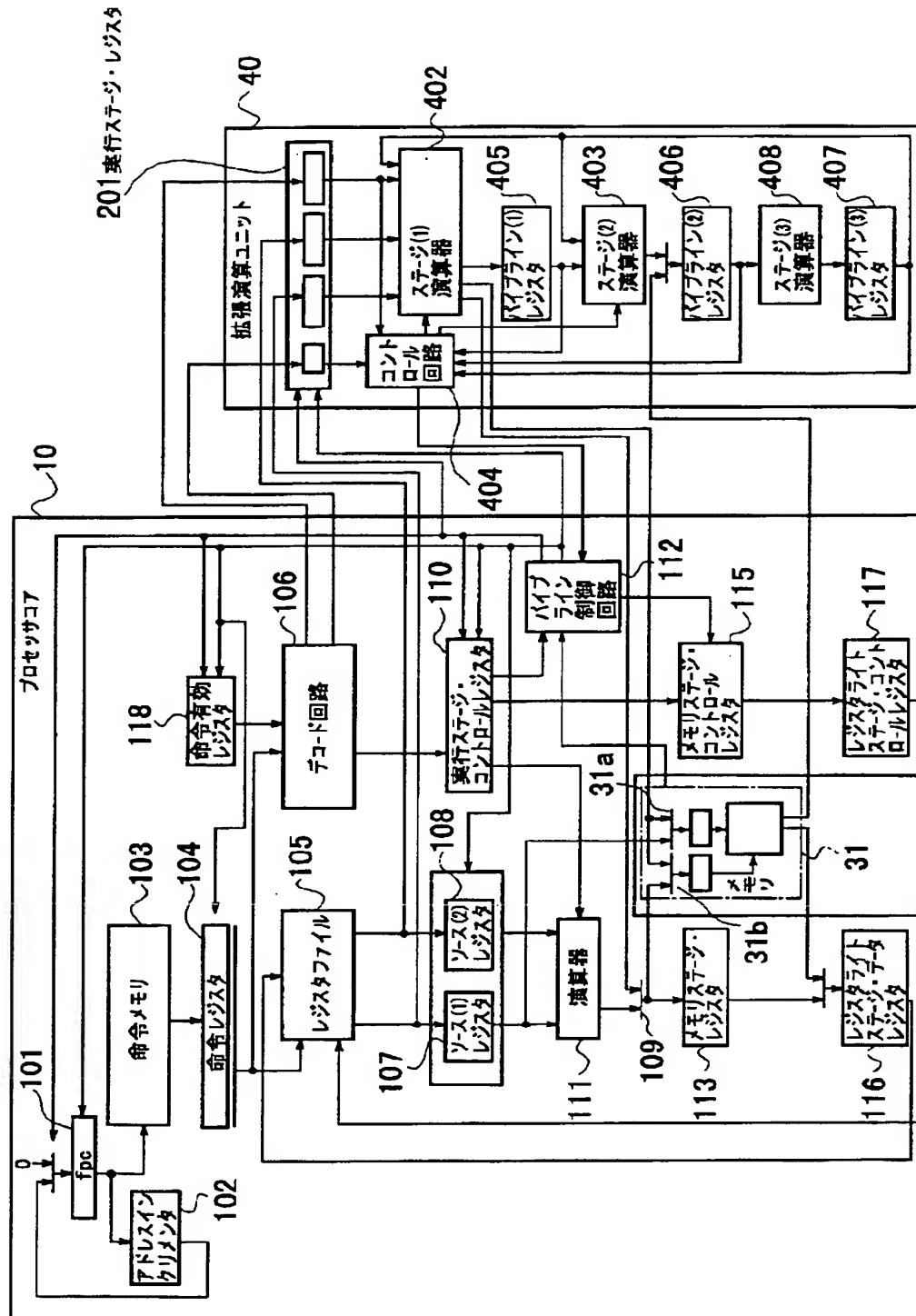
【図 4】

演算器制御入力	演算器出力
00 (nop)	ソースレジスタ107
01 (add)	ソースレジスタ107+ソースレジスタ108
10 (ext)	拡張演算ユニットからの入力

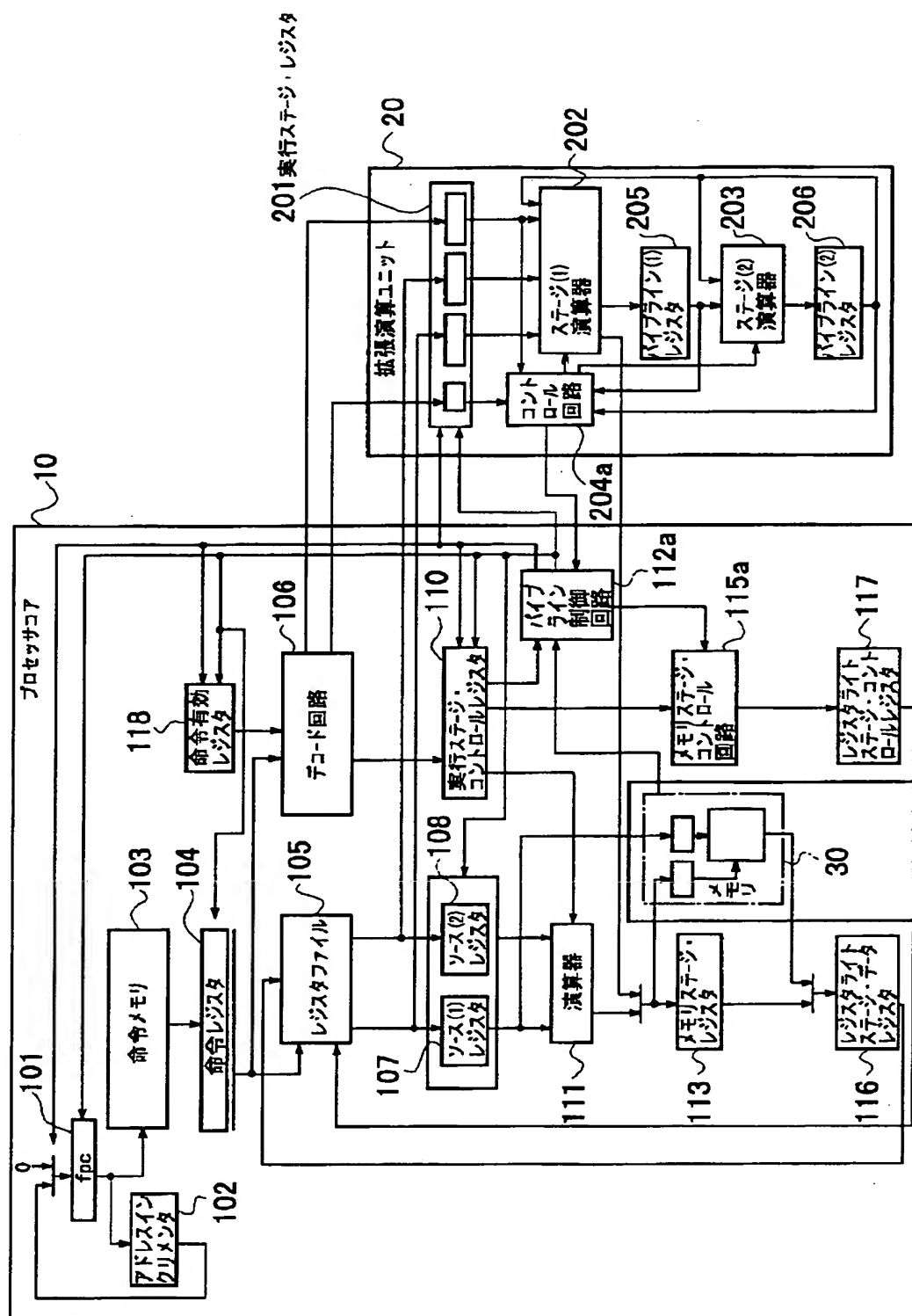
【图 1】



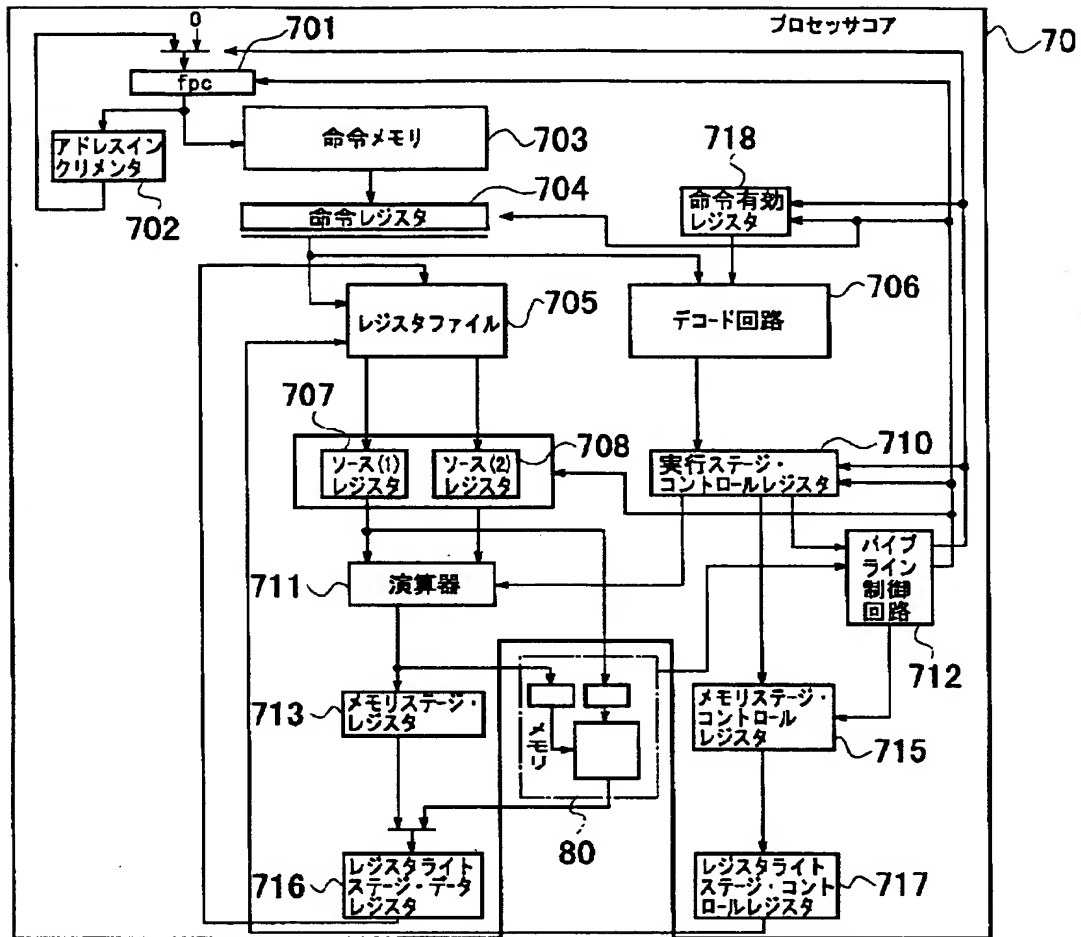
201 実行ステージ・レジスタ



【図 6】



【図7】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.